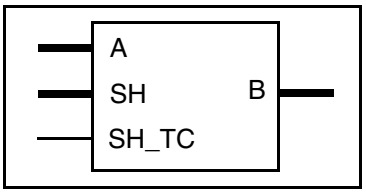
DW_sla

Arithmetic Shifter with Preferred Left Direction (VHDL style)

Version, STAR and Download Information: IP Directory

Features and Benefits

- Parameterized data and shift coefficient word lengths
- Uses VHDL semantics for the arithmetic shift operation
- Capable of shifting in both directions



Revision History

Description

DW_sla is an arithmetic shifter that has the same semantics as the sla operator in VHDL. A list of input and output pins is shown in Table 1-1. The component has the parameters shown in Table 1-2 on page 2. The parameters control the number of bits used in the component's ports.

The DW_sla may be configured to work as a bidirectional or unidirectional shifter. The SH_TC input indicates if the shifting distance (SH) is positive or negative. When SH_TC = 1 the input SH is interpreted as a signed integer represented in two's complement. The component has a preferred left direction for shifting, which means that when SH > 0 the left shift operation is performed. The input data A is always shifted to the left when SH_TC = 0 (the SH input value is always positive) or when SH_TC = 1 and SH > 0 (the SH input is signed and positive). Otherwise, input A is shifted to the right by ($-2^{SH\_width}+SH_{rep}$) bits, where SH_{rep} is the unsigned integer value of SH.

The arithmetic right shift operation is executed the same way as other arithmetic shifters, the MS bit is copied to all positions that are made open. Differently from other shifters, when shifting to the left, the LS bit is copied to all the positions that are made open. For example, when SH_TC = 0, SH = 2, and $A=(100101)_2$, the output B is $(010111)_2$, where the LS bit of A was replicated on the rightmost positions during the left shift operation. Table 1-5 on page 2 illustrates other combinations of the input values for a small component.

Table 1-1 Pin Description

Pin Name	Width	Direction	Function
A	*A_width*	Input	Input data
SH	*SH_width*	Input	Shift control
SH_TC	1 bit	Input	Shift two's complement control ■ 0 = Unsigned ■ 1 = Signed
B	*A_width*	Output	Shifted data out

Table 1-2 Parameter Description

Parameter	Values	Description
A_width	≥ 2	Word length of `A` and `B`
SH_width	≥ 1	Word length of `SH`

Table 1-3 Synthesis Implementations[a]

Implementation Name	Function	License Feature Required
str	Synthesis model targeted for speed	DesignWare
astr	Synthesis model targeted for area	DesignWare
mx2	Static implement using 2:1 multiplexers only.	DesignWare

a. During synthesis, Design Compiler selects the appropriate architecture for your constraints. However, you may force Design Compiler to use one of the architectures described in this table.

Table 1-4 Simulation Models

Model	Function
dw/dw01/src/DW_sla_sim.vhd	VHDL simulation model source code
dw/sim_ver/DW_sla.v	Verilog simulation model source code

The following truth table specifies behavior with *A_width* = 8, *SH_width* = 3.

Table 1-5 Truth Table (*A_width* = 8, *SH_width* = 3)

SH(2:0)	SH_TC	B(7)	B(6)	B(5)	B(4)	B(3)	B(2)	B(1)	B(0)
000	X	A(7)	A(6)	A(5)	A(4)	A(3)	A(2)	A(1)	A(0)
001	X	A(6)	A(5)	A(4)	A(3)	A(2)	A(1)	A(0)	A(0)
010	X	A(5)	A(4)	A(3)	A(2)	A(1)	A(0)	A(0)	A(6)
011	X	A(4)	A(3)	A(2)	A(1)	A(0)	A(0)	A(0)	A(0)
100	0	A(3)	A(2)	A(1)	A(0)	A(0)	A(0)	A(0)	A(0)
101	0	A(2)	A(1)	A(0)	A(0)	A(0)	A(0)	A(0)	A(0)
110	0	A(1)	A(0)	A(0)	A(0)	A(0)	A(0)	A(0)	A(0)
111	0	A(0)	A(0)	A(0)	A(0)	A(0)	A(0)	A(0)	A(0)
100	1	A(7)	A(7)	A(7)	A(7)	A(7)	A(6)	A(5)	A(4)
101	1	A(7)	A(7)	A(7)	A(7)	A(6)	A(5)	A(4)	A(3)
110	1	A(7)	A(7)	A(7)	A(6)	A(5)	A(4)	A(3)	A(2)

Table 1-5 Truth Table (*A_width* = 8, *SH_width* = 3) (Continued)

SH(2:0)	SH_TC	B(7)	B(6)	B(5)	B(4)	B(3)	B(2)	B(1)	B(0)
111	1	A(7)	A(7)	A(6)	A(5)	A(4)	A(3)	A(2)	A(1)

Related Topics

- Logic – Combinational Overview
- DesignWare Building Block IP User Guide

HDL Usage Through Component Instantiation - VHDL

```
library IEEE,DWARE;
use IEEE.std_logic_1164.all;
use DWARE.DWpackages.all;
use DWARE.DW_Foundation_comp.all;

entity DW_sla_inst is
      generic (
        inst_A_width : POSITIVE := 8;
        inst_SH_width : POSITIVE := 3
        );
      port (
        inst_A : in std_logic_vector(inst_A_width-1 downto 0);
        inst_SH : in std_logic_vector(inst_SH_width-1 downto 0);
        inst_SH_TC : in std_logic;
        B_inst : out std_logic_vector(inst_A_width-1 downto 0)
        );
    end DW_sla_inst;


architecture inst of DW_sla_inst is

begin

    -- Instance of DW_sla
    U1 : DW_sla
    generic map ( A_width => inst_A_width, SH_width => inst_SH_width )
    port map ( A => inst_A, SH => inst_SH, SH_TC => inst_SH_TC, B => B_inst );


end inst;

-- pragma translate_off
configuration DW_sla_inst_cfg_inst of DW_sla_inst is
for inst
end for; -- inst
end DW_sla_inst_cfg_inst;
-- pragma translate_on
```

HDL Usage Through Component Instantiation - Verilog

```
module DW_sla_inst( inst_A, inst_SH, inst_SH_TC, B_inst );

parameter A_width = 8;
parameter SH_width = 3;


input [A_width-1 : 0] inst_A;
input [SH_width-1 : 0] inst_SH;
input inst_SH_TC;
output [A_width-1 : 0] B_inst;

    // Instance of DW_sla
    DW_sla #(A_width, SH_width)
      U1 ( .A(inst_A), .SH(inst_SH), .SH_TC(inst_SH_TC), .B(B_inst) );

endmodule
```

Revision History

For notes about this release, see the *DesignWare Building Block IP Release Notes*.

For lists of both known and fixed issues for this component, refer to the STAR report.

For a version of this datasheet with visible change bars, click here.

Date	Release	Updates
September 2018	DWBB_201806.2	■ Updated example in "HDL Usage Through Component Instantiation - VHDL" on page 4 ■ Added this Revision History table and the document links on this page

Copyright Notice and Proprietary Information

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at https://www.synopsys.com/company/legal/trademarks-brands.html.

All other product or company names may be trademarks of their respective owners.

Free and Open-Source Software Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.
www.synopsys.com